IN THE CLAIMS:

1. (PREVIOUSLY PRESENTED) A method for a first file server to provide file service operations normally performed by a second file server after the second file server suffers an error condition, the first and second file servers operatively interconnected with a set of clients using a network protocol, the network protocol being free of support for moving a transport address from the second file server to the first file server, the method comprising the steps of:

detecting, by the first file server, that the second file server has suffered an error condition;

asserting ownership, by the first file server, of a set of storage devices normally owned by the second file server;

activating, on the first file server, a secondary data access port for receiving connections over a network; and

processing, by the first file server, file service operations directed to the secondary data access port from a set of failover clients, the failover clients accessing the first file server by computing a network address associated with the first file server from a symbolic name generated from the second file server, whereby failover operation is achieved by the client.

2. (ORIGINAL) The method of claim 1 wherein the step of detecting the error condition further comprises the steps of sending, by the second file server, an error message to the first file server.

3. (ORIGINAL) The method of claim 1 wherein the step of detecting an error condition further comprises the step of:

detecting, by the first file server, a lack of a status signal generated by the second file server.

4. (ORIGINAL) The method of claim 1 wherein the secondary data access port is a virtual interface discriminator.

5. (ORIGINAL) A method for a client to continue to access file service operations after a first file server has suffered an error condition, the method comprising the steps of:

computing a failover name;

resolving the failover name to a network address; and

connecting to a failover file server using the network address and a predetermined alternate data access port.

6. (PREVIOUSLY PRESENTED) A method for a client to continue to access file service operations after a first file server has suffered an error condition, the method comprising the steps of:

computing a failover name by appending a set text string to a name of the first file server;

resolving the failover name to a network address;

connecting to a failover file server using the network address and a predetermined alternate data access port.

7. (ORIGINAL) The method of claim 5 wherein the predetermined alternate data access port further comprises a virtual interface discriminator.

8. (ORIGINAL) A file server for use in a file server cluster, the file server operatively interconnected with a set of clients using a network protocol, the network protocol being free of support for moving a transport address from a first file server to a second file server, the file server comprising:

a cluster interconnect, the cluster interconnect providing a communications link to a partner file server in the file server cluster;

a primary data access port for receiving file service operations from file server clients; and

a secondary data access port, the secondary data access port only being active when the file server detects that the partner file server has suffered an error condition, wherein the file server processes file service operations received via the secondary data access port to provide file service operations to clients of the partner file server.

9. (ORIGINAL) The file server of claim 8 wherein the primary data access port further comprises a virtual interface discriminator.

10. (ORIGINAL) The file server of claim 9 wherein the secondary data access port further comprises a virtual interface discriminator.

11. (ORIGINAL) A file server for use in a file server cluster, the file server operatively interconnected with a set of clients using a network protocol, the network protocol being free of support for moving a transport address from a first file server to a second file server, the file server comprising:

means for communicating with a partner file server in the file server cluster;

means for identifying that the partner file server has suffered an error condition;

means asserting ownership of disks normally owned by the partner file server; and

means for processing file service operations from clients of the partner file server.

12. (ORIGINAL) A computer-readable medium, including program instructions executing on a file server, for providing file service operations normally performed by a failed file server, the program instructions performing the steps of:

detecting that the failed file server has suffered an error condition;

asserting ownership of a set of storage devices normally owned by the failed file server;

activating a secondary data access port for receiving connections over a network; and

processing file service operations received by one or more clients over the data access port.

13. (ORIGINAL) A computer-readable medium, including program instructions executing one client, for the client to continue to access file service operations after a first file server has suffered an error condition, the instructions including steps for:

computing a failover name;

resolving the failover name to a network address; and

connecting to a failover file server using the network address and a predetermined alternate data access port.

14. (CURRENTLY AMENDED) A method for operating a computer failover system, comprising:

executing a client computer program on a client computer, the client computer program communicating with a first file server;

computing, by a ~~fie~~ file system process communicating with the client computer program, a failover name;

resolving the failover name to a network address;

detecting an error condition; and

connecting, in response to detecting the error condition, to a failover file server port having the network address.

15. (PREVIOUSLY PRESENTED) The method as in claim 14, further comprising:

using a file server name for communicating with the first file server; and

computing the failover name by modifying the file server name by an alphanumeric text.

16. (PREVIOUSLY PRESENTED) The method as in claim 14, further comprising:

computing the failover name by appending the text "backup" to a file server name used to communicate with the first file server.

17. (PREVIOUSLY PRESENTED) The method as in claim 14, further comprising:

transmitting the failover name to a distributed naming service to perform the step of resolving the failover name to a network address.

18. (PREVIOUSLY PRESENTED) The method as in claim 14, further comprising:

using a database program as the client computer program.

19. (CURRENTLY AMENDED) The method as in claim 14, wherein the step of detecting the error condition further comprises:

detecting a lack of a heartbeat signal from ~~the~~ a failed file server.

20. (CURRENTLY AMENDED) The method as in claim 14, wherein the step of detecting the error condition further comprises:

transmitting by ~~the~~ a failing file server an "I am failing" message.

21. (CANCELLED)

22. (CURRENTLY AMENDED) A computer failover system, comprising:

means for executing a client computer program on a client computer, the client computer program communicating with a first file server;

means for computing, by a ~~fie~~ file system process communicating with the client computer program, a failover name;

means for resolving the failover name to a network address;

means for detecting an error condition; and

means for connecting, in response to detecting the error condition, to a failover file server port having the network address.

23. (PREVIOUSLY PRESENTED) The system as in claim 22, further comprising:

means for using a file server name for communicating with the first file server; and

means for computing the failover name by modifying the file server name by an alphanumeric text.

24. (PREVIOUSLY PRESENTED) The system as in claim 22, further comprising:

means for computing the failover name by appending the text "backup" to a file server name used to communicate with the first file server.

25. (PREVIOUSLY PRESENTED) The system as in claim 22, further comprising:

means for transmitting the failover name to a distributed naming service to perform the step of resolving the failover name to a network address.

26. (PREVIOUSLY PRESENTED) The system as in claim 22, further comprising:

means for using a database program as the client computer program.

27. (CURRENTLY AMENDED) The system as in claim 22, further comprising:

means for detecting a lack of a heartbeat signal from ~~the~~ a failed file server.

28. (CURRENTLY AMENDED) The system as in claim 22, further comprising:

means for sending, by ~~the~~ a failing file server, an error message to the first file server.

29. (PREVIOUSLY PRESENTED) The system as in claim 22, further comprising:

means for transmitting by the failing file server an "I am failing" message.

30. (CURRENTLY AMENDED) A computer failover system, comprising:

a client computer having a client computer program executing thereon, the client computer program communicating with a first file server;

a ~~fie~~ file system process communicating with the client computer program, the file system process computing a failover name;

a port to transmit the failover name to a distributed name server to resolve the failover name to a network address;

a port to receive a message reporting an error condition in the first file server; and

a file system process to use the failover name and network address to connect, in response to the error condition, to a failover file server port having the network address.

31. (PREVIOUSLY PRESENTED) The system as in claim 30, further comprising:

a file system process to use a file server name to communicate with the first file server, and to compute the failover name by modifying the file server name by an alpha-numeric text.

32. (PREVIOUSLY PRESENTED) The system as in claim 30, further comprising:

a file system process to compute the failover name by appending the text "backup" to a file server name used to communicate with the first file server.

33. (PREVIOUSLY PRESENTED) The system as in claim 30, further comprising:

a file system process to transmit the failover name to a distributed naming service to perform the step of resolving the failover name to a network address.

34. (PREVIOUSLY PRESENTED) The system as in claim 30, further comprising:

the client computer program is a database program.

35. (CURRENTLY AMENDED) The system as in claim 30, further comprising:

means for detecting a lack of a heartbeat signal from ~~the~~ a failed file server.

36. (CURRENTLY AMENDED) The system as in claim 30, further comprising:

means for sending, by ~~the~~ a failing file server, an error message to the first file server.

37. (PREVIOUSLY PRESENTED) The system as in claim 30, further comprising:

means for transmitting by the failing file server an "I am failing" message.

38. (CURRENTLY AMENDED) A computer readable media, comprising:

said computer readable media containing instructions for execution on a processor for the practice of a method for operating a computer failover system, the method having the steps of,

executing a client computer program on a client computer, the client computer program communicating with a first file server;

computing, by a ~~fie~~ file system process communicating with the client computer program, a failover name;

resolving the failover name to a network address;

detecting an error condition; and

connecting, in response to detecting the error condition, to a failover file server port having the network address.

39. (CURRENTLY AMENDED) Electromagnetic signals propagating on a computer network, comprising:

said electromagnetic signals carrying instructions for execution on a processor for the practice of a method for operating a computer failover system, the method having the steps of,

executing a client computer program on a client computer, the client computer program communicating with a first file server;

computing, by a ~~fie~~ file system process communicating with the client computer program, a failover name;

resolving the failover name to a network address;

detecting an error condition; and

connecting, in response to detecting the error condition, to a failover file server port having the network address.